Attorney Docket No.: 15065US01

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the Application of:

Wade Keith Wan, et al.

Serial No.: 10/642,318

Filed: August 15, 2003

For: PSEUDO-RANDOM NUMBER GENERATION BASED ON PERIODIC SAMPLING OF ONE OR MORE LINEAR FEEDBACK SHIFT REGISTERS

Examiner: Eleni A. Shiferaw

Group Art Unit: 2436

Conf. No.: 2849

Electronically Filed on February 24, 2009

REPLY BRIEF

Board of Patent Appeals and Interferences
U.S. Patent and Trademark Office
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with 37 CFR 41.41, the Appellants respectfully submit this Reply Brief in response to the Examiner's Answer mailed on December 24, 2008. This Reply Brief provides a timely response to the Examiner's Answer and has a period of reply that expires on February 24, 2009.

STATUS OF THE CLAIMS

The present Application originally included 24 claims (Claims 1-24). Claims 23-24 were withdrawn. Pending Claims 1-22 stand rejected and are the subject of this appeal.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

I. Claims 1, 3-6, 14-16, and 20-22 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Publication No. 2004/0205095 (hereinafter, Gressel).

II. Claim 1 stands rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,993,542 (hereinafter, Meiyappan).

III. Claims 7-10 and 19 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 5,327,522 (hereinafter, Furuta).

IV. Claims 11-13 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Publication No. 2003/0072059 (hereinafter, Thomas).

V. Claim 17 stands rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Publication No. 2005/0066168 (hereinafter, Walmsley).

VI. Claim 18 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Furuta in view of Gressel.

APPLICANT'S RESPONSE TO EXAMINER'S RESPONSE TO ARGUMENT

The Appellants maintain the arguments previously presented in the Brief on Appeal. In the Grounds of Rejection section (section 9) of the Examiner's Answer, the Examiner maintains the arguments she previously presented in the final Office Action. However, the Examiner has presented new arguments in Examiner's "Response to Argument" (section 10) in response to the Brief on Appeal, which were not made previously in the final Office Action. Therefore, the Appellants believe that the Examiner has made some new grounds of rejection. As a consequence, the Appellants have addressed these newly presented arguments in this Reply Brief. The Appellants question if the Examiner has received approval from the Technology Center Director to introduce these new grounds of rejection in this Examiner's Answer. The Appellants believe that new grounds of rejection have been made because the Examiner's brings up new arguments (along with new support from the cited reference(s)) in which the Appellants had not been given an opportunity to react to the rejection. The Appellants have denoted Examiner's newly presented arguments/ remarks in bold text. The Appellants have responded to these new arguments in this reply brief.

This Reply Brief responds to the Examiner's "Response to Argument" (section 10) starting from page 7 of the Examiner's Answer. The Appellants believe that the pending claims recite patentable subject matter. Consequently, the Appellants respectfully submit that the Board should reverse the rejections to Claims 1-22.

I. REJECTION OF CLAIMS 1, 3-6, 14-16, AND 20-22 UNDER 35 U.S.C. § 102(e)

With respect to the rejections to Claims 1, 3-6, 14-16, and 20-22, the Appellants maintain the arguments previously presented in the Brief on Appeal.

A. Independent Claim 1

Claim 1 is directed to:

1. A method of generating pseudo-random numbers using a linear feedback shift register in which the correlation between successive pseudo-random numbers is reduced, said method comprising sampling output sequences of said linear feedback shift register with a specified periodicity.

Regarding Claim 1, the Examiner's Answer states:

> Regarding argument Gressel failure [sic] to disclose "sampling output sequences of linear feedback shift register with a specified periodicity," appeal brief pages 6-7, argument is not persuasive because Gressel discloses generating a cyclic output sequence of pseudorandom binary numbers wherein the cyclic output sequence including a basic sequence which is generated repeatedly, using clocked pseudorandom binary number sequence generator (see abstract/par. 0080, and par. 0049).
>
> The examiner interpreted the "specified periodicity" of claim 1 in light of the appellant's disclosure par. 0024 lines 6-8 wherein "A Linear Feedback Shift Register (LFSR) of any given size n is capable of producing every possible state (outcome) during the period [p=(2^n)-1]." Further, see par 0028 lines 1-4 for specified n bit/digit output (n=3). See par. 0029 lines 3-4 of appellant's disclosure that discloses "sampling once every n iterations prevents revealing ....." The "specified periodicity" of the claim is interpreted as a period/interval time/clock cycle. **The applied reference Gressel teaches the LFSR register sampling randomly once in 64 system clock cycles see par. 0049.** The clocked LFSR of Gressel generates a cyclic output sequence of binary numbers (see 0026-0027 and abstract).
>
> Regarding argument Gressel teaching [sic] sampling "after a random waiting interval has elapsed" which is opposite from sampling with a "specified

> periodicity," appeal brief page 8, argument is not persuasive because see par. [0047-**0049**] wherein Gressel discloses sampling randomly and not more often than once in 64 system clock cycles (see par. [0047-**0049**]) **and further see par. [0267, and 0175-0177] wherein Gressel discloses a clocked binary random number generator wherein the generator is an electronic oscillator that generates periodic signal for synchronization of processes and the randomness of the generator is typically initiated by simultaneously activating a primary clock and a second uncorrelated clock, such that randomizing events occur at intractably difficult to estimate intervals and the pseudorandom modification includes a random slip in which a portion of the cyclic output sequence is omitted (see par. 0088).** Therefore Gressel does not teach away from what is recited in claim 1.
>
> *See* Examiner's Answer at pages 8-9.

The Appellants respectfully submit that the Examiner has brought up a new ground of rejection when she refers to Gressel, at paragraph [0049]. Claim 1 recites "a method of generating pseudo-random numbers using a linear feedback shift register in which the correlation between successive pseudo-random numbers is reduced, said method comprising sampling output sequences of said linear feedback shift register with a specified periodicity." The Appellants respectfully maintain that the Examiner has not shown a teaching of Claim 1. For example, the Examiner has not shown a teaching of "said method comprising sampling output sequences of said linear feedback shift register with a specified periodicity." The Examiner refers to the Appellants' disclosure at paragraph [24]. While paragraph [24] discloses that "a LFSR of any given size n (corresponding to the number of registers in the LFSR) is capable of producing every possible state (except the all zero state) during the period p=2n-1, but will do so only if one or more feedback taps are properly configured, as will be discussed shortly," paragraph [0024] does not provide any support for the Examiner in her attempt to show a teaching of "sampling output sequences of said linear feedback shift register with a specified periodicity." However, the Examiner may refer to paragraph [24] which discloses how an LFSR operates. Furthermore, the Appellants' disclosure, at paragraph [28], discloses sampling the LFSR output at n=3 iterations. Thus, paragraph [28] provides an embodiment in which an LFSR output is sampled at every three iterations. Furthermore, if the Examiner has correctly interpreted Claim 1 in light of the specification, it should be clear that Gressel does not teach anything about "sampling output sequences of said linear feedback shift register with a specified periodicity."

The Examiner references Gressel, at paragraph [0049], which states:

> If the feedback bit is two's complemented (XORed) with a random "1" and shifted into the leftmost flip-flop, the contents are altered to a stage "forward" which might "normally occur" an equiprobable natural number, smaller than 2^ n, of clock cycles later, as illustrated and explained herein. According to a preferred embodiment of the present invention, sampling is typically enacted only randomly and preferably not more often than once in 64 system clock cycles.

The Examiner newly argues that Gressel, at paragraph [0049], teaches "sampling output sequences of said linear feedback shift register with a specified periodicity." The Examiner states that "Gressel teaches the LFSR register sampling randomly once in 64 system clock cycles." The Appellants respectfully submit that the Examiner has mischaracterized what is actually stated in Gressel. Gressel, at paragraph [0049], actually states that "according to a preferred embodiment of the present invention, sampling is typically *enacted only randomly* and preferably not more often than once in 64 system clock cycles." Thus, based on the foregoing, Gressel's preferred embodiment is such that "sampling is typically enacted *only randomly.*" Thus, for at least this reason, Gressel, at paragraph [0049], does not teach "sampling output sequences of said linear feedback shift register *with a specified periodicity*," as recited in Claim 1. Furthermore, Gressel, at paragraph [0049], teaches that this random sampling is preferably performed not more often than once in 64 system clock cycles. This means that the random sampling occurs randomly at a periodicity which is greater than or equal to 64 clock cycles. For example, the first sample may occur at the 64th clock cycle, the second sample may occur at the 69th clock cycle, the third sample may occur at the 89th clock cycle, and so on. Thus, Gressel teaches random sampling of a linear feedback shift register. Thus, for at least these reasons, Gressel, at paragraph [0049], does not teach "sampling output sequences of said linear feedback shift register with a specified periodicity," as recited in Claim 1. Therefore, Claim 1 contains patentable subject matter that should be passed to allowance.

The Examiner references Gressel, at paragraph [0267], which states:

> FIG. 3A is a simplified functional block diagram of microelectronic apparatus 500 for generating binary words. The apparatus of FIG. 3A preferably comprises at least one clocked pseudorandom binary number sequence generator normally operative to generate a cyclic output sequence of binary numbers. Both random slips and random swaps occasionally occur in the cyclic output sequence thereby altering the output sequence. The apparatus of FIG. 3A preferably implements a combined randomization procedure of the LFSR enhancements of FIGS. 1A and 2. Inputs 520 and 510 are operative to enact the random slip and the random swap, respectively. Input 560 clocks flip-flops FF1 to FF5 in shift register 542. Subject to random occurrences of slip pulses, on line 520 and random swapping of the feedback configuration caused by toggled inputs on line 510, the output on line 550 is a 5 bit pseudorandom binary word. The apparatus generates an LFSR output 570, for either of the two random swap configurations, a

feedback 580 XORed to the random slip pulse, an output 530 to prevent a "stuck on zero" syndrome and an output 540 to prevent a "long run of one" syndrome.

Furthermore, the Examiner references Gressel, at paragraphs [0175-0177], which states:

> Clock: The device, typically an electronic oscillator that generates periodic signals for synchronization of processes. In both preferred embodiments, randomness is typically initiated by simultaneously activating a primary clock, also termed herein a "system clock", and a second uncorrelated clock, such that randomizing events occur at intractably difficult to estimate intervals. A typical clock cycle occupies a time interval, called a period. Typically, during the majority of the first half of the period the clock cycle signal is stable at a binary one voltage, and during the majority of the second half of the clock period, the voltage is stable at a binary zero level.
>
> Clock Modes: Two clock modes are described: Single clock mode and dual clock mode. In single clock mode, only a primary clock, e.g. primary clock 1040 in FIG. 10, is operative. Primary clock 1040, when operative, typically activates all nLFSRs in the random number generator. In dual clock mode, both a primary clock and an additional, slower, uncorrelated clock, derived from an oscillator typically uncorrelated to the primary clock, are operative. Clock 1030 in FIG. 10 is an example of a slow or uncorrelated clock. Clock 1030, when operative, typically forces all nLFSRs in the random number generator into an unpredictable condition.
>
> Either of these clock modes may be operative in each of the random number generators shown and described herein such as the random number generators of FIGS. 6 and 10. In the illustrated embodiments, the dual clock mode is employed in the random number generator of FIG. 6 and both modes are employed in the random number generator of FIG. 10. Typically, a primary clock is enabled for all operations between autonomous devices. This prevents glitches and metastable oscillations between devices within the random number generator and between the random number generator and the host device. In the random

> number generator of FIG. 6, delays and decelerated operation of the device utilize the Slow Clock 1030.

The Examiner alleges that Gressel, at paragraphs [0267] and [0175-0177], discloses a clocked binary random number generator wherein the generator is an electronic oscillator that generates periodic signal for synchronization of processes and the randomness of the generator is typically initiated by simultaneously activating a primary clock and a second uncorrelated clock, such that randomizing events occur at intractably difficult to estimate intervals and the pseudorandom modification includes a random slip in which a portion of the cyclic output sequence is omitted (see par. 0088)." Instead, Gressel, at paragraph [0267], discloses a "clocked pseudorandom binary number sequence generator normally operative to generate a cyclic output sequence of binary numbers," in which "random slips and random swaps occasionally occur in the cyclic output sequence thereby altering the output sequence." Thus, by way of incorporating such "random slips" and "random swaps" in a cyclic output sequence, the pseudorandom sequence is generated. The Appellants respectfully submit that incorporating such random slips and random swaps does not teach "sampling output sequences of said linear feedback shift register with a specified periodicity," as recited in Claim 1. For example, Gressel teaches using random swaps in a cyclic output sequence as a way to generate pseudorandom binary numbers, which does not teach "sampling output sequences of said linear feedback shift register with a specified periodicity." Thus, for at least these reasons, Claim 1 contains patentable subject matter, that should be allowed. Furthermore, contrary to what the Examiner alleges, Gressel, at paragraphs [0175-0177], discloses a clock and its modes, which may be used to clock a feedback shift register. Therefore, the clock described by Gressel, at paragraphs [0175-0177], does not disclose anything about "sampling output sequences of [a] linear feedback shift register with a specified periodicity," as recited in Claim 1. Thus, Gressel describes modifying a clock used for clocking a linear feedback shift register or incorporating a random slip or a random swap in the output of a linear feedback shift register. Thus, modifying a clock input or incorporating random slips or swaps does not teach anything about "sampling output sequences of said linear feedback shift register with a specified periodicity."

Therefore, based on the foregoing Appellants' arguments, the Appellants respectfully submit that the Examiner has not shown a teaching of each and every element recited in Claim 1. Consequently, Claim 1 contains patentable subject matter. The Appellants request a reversal of the rejection to Claim 1. Furthermore, for at least the reason that Claims 2-6, 14-16, and 20-22 depend on independent Claim 1, Claims 2-6, 14-16, and 20-22 should be passed to allowance as well.

B. Dependent Claim 3

Claim 3 is directed to:

3. The method of Claim 1 wherein said specified periodicity is equal to the number of bits output by said linear feedback shift register.

Regarding Claim 3, the Examiner's Answer states:

> Regarding argument the applicant could not see how Gressel, at paragraph [0175], shows a teaching of "wherein said specified periodicity is equal to the number of bits output by said linear feedback shift register," appeal brief page 9, argument is not persuasive because Gressel par. [0175] **teaches the device (the clocked binary random number generator) generating and outputting**

> **random number bits periodically in a time interval and/or Gressel further teaches the clocked pseudorandom binary number sequence generator includes a feedback shift register and wherein the pseudorandom displacement is caused by complementing the serial feedback bit in the feedback shift register using pulsed "1" bits which are externally generated at intractably difficult to estimate intervals of time (see par. [0084]).**
>
> *See* Examiner's Answer at page 9.

The Examiner references Gressel, at paragraph [0175], which states:

> Clock: The device, typically an electronic oscillator that generates periodic signals for synchronization of processes. In both preferred embodiments, randomness is typically initiated by simultaneously activating a primary clock, also termed herein a "system clock", and a second uncorrelated clock, such that randomizing events occur at intractably difficult to estimate intervals. A typical clock cycle occupies a time interval, called a period. Typically, during the majority of the first half of the period the clock cycle signal is stable at a binary one voltage, and during the majority of the second half of the clock period, the voltage is stable at a binary zero level.

Furthermore, the Examiner references Gressel, at paragraph [0084], which states:

> Further in accordance with a preferred embodiment of the present invention, the clocked pseudorandom binary number sequence generator includes a feedback shift register and wherein the pseudorandom displacement is caused by complementing the serial feedback bit in the feedback shift register using pulsed "1" bits which are externally generated at intractably difficult to estimate intervals of time.

As was stated in the Brief on Appeal, the Appellants respectfully submit that Gressel, at paragraph [0175], teaches how a "clock" operates and functions to generate periodic signals for synchronization. This has nothing to do with “wherein said specified periodicity is equal to the number of bits output by said linear feedback shift register,” as recited in Claim 3. Consequently, for at least this reason, the Examiner has not shown a teaching of Claim 3. Furthermore, the Examiner has newly introduced Gressel, at paragraph [0084], in an attempt to show a teaching of Claim 3. While Gressel, at paragraph [0084], discloses how “pseudorandom displacement is caused by complementing the serial feedback bit in the feedback shift register using pulsed "1" bits which are externally generated at intractably difficult to estimate intervals of time,” Gressel, at paragraph [0084], does not disclose anything about “wherein said specified periodicity is equal to the number of bits output by said linear feedback shift register,” as recited in Claim 3. Based on the foregoing reasons, the Appellants respectfully submit that the Examiner has not shown a teaching of each and every element recited in Claim 3. Consequently, Claim 3 contains patentable subject matter. The Appellants request a reversal of the rejection to Claim 3.

C. Dependent Claims 4-6

Using Claim 4 as exemplary for Claims 4-6, Claim 4 is directed to:

4. The method of Claim 1 further comprising periodically switching between iterative outputs generated by two or more linear feedback shift registers.

Regarding Claims 4-6, the Examiner’s Answer states:

> Appellant's argument regarding Gressel failure to teach "periodically switching between iterative outputs generated by two or more linear feedback shift registers," appeal brief pages 10-12, is not persuasive because Gressel on

par. [0262-0264] teaches alternating/swapping output between two feedback configurations (see fig. 10) that discloses two nLFSRs 1200 and 1300. Switching/swapping feedback tap configurations (feedback swaps) of the two feedback shift registers 1200 and 1300 are described in (par. [0293]) and further Gressel (par. [0244]) teaches swapping for randomizing the output of LFSR between two sets of feedback taps (configurations) hence, causing alternating generation of cyclic/periodic segments from two maximum length linear feedback register sequences. Par. [0281-0282] of Gressel discloses randomizing the three non-linear feedback shift registers nLFSRs 640, 650 and 660 by using slip trigger generator to switch in a regular interval.
See Examiner's Answer at pages 9-10.

The Examiner references Gressel, at paragraphs [0262-0264], which states:

> FIG. 2 is a schematic representation of another length 5 random enhancing modification of a conventional LFSR. The same 5 celled LFSR of FIG. IA is converted into a non-linear feedback shift register device, using a second enhancement used in the preferred embodiment of FIG. 10. The device of FIG. 2 demonstrates the swap sequence configuration enacted by randomly alternating the device between one feedback configuration to a second feedback configuration.
>
> The two feedback configurations include: (a) a first configuration with shift register 442 output taps only from flip-flops FF2 and FF5, these output taps also termed herein "feedbacks 2 and 5"; and (b) a second configuration, wherein feedbacks from flip-flops FF3 and FF4 are complemented (added to) feedbacks 2 and 5 by a binary one-enabling input on line 410.
>
> When Random Swap Select on line 410 is a one, AND gate 470 switches in the feedback output from flip-flops FF3 and FF4, XORed in exclusive or gate 447, into the results of the output of AND gate, 447. In this four tap feedback configuration, the output from XOR gate 447 is XOR'd by XOR gate 449, to the feedbacks from flip-flops FF2 and FF5. The random swap select on line 410, therefore, transforms the device to a configuration with a single pair feed back to

> a double pair feedback. The device alternates between one configuration and the other, as the signal on line 410 oscillates.

Furthermore, the Examiner references Gressel, at paragraphs [0281-0282], which states:

> The binary contents of each of the nLFSRs 640, 650 and 660 is randomized by two uncorrelated sources. The slip triggers, on lines 622, 624 and 636, emanating from slip trigger generator 670 at staggered instants from slip trigger bus 671, emanate at regular intervals switched in turn in regular intervals, regulated by the fast clock. The average random sequence slip displacement at such triggers is 2^ n/4, where n is the number of flip-flops in the nLFSR register. The second source of unpredictability, inherent to each nLFSR, is the change of frequencies of the driving clocks on lines 642, 652 and 662.
>
> Responsive to each slip trigger command, a corresponding Slip & Mixed Clock Generator 643, 653 or 663 switches the frequency on its corresponding clock line 642, 652 or 662, from the fast clock to the slow clock, for a random interval (a random number of slow clock cycles), as prescribed in the flowchart of FIG. 8A for nLFSR 640. The process described in the flowchart of FIG. 8A for nLFSR 640 may be identical to the random deceleration in nLFSRs 650 and 660. Preferred synchronized timing of the random decelerated clocks generated by clock generators 643, 653 and 663, to avoid glitches, is illustrated in the timing diagram of FIG. 9.

The Appellants respectfully submit that Gressel, at paragraphs [0262-0264], does not teach anything about "*periodically switching* between iterative outputs generated by two or more linear feedback shift registers." Contrary to what the Examiner alleges, Gressel, at paragraph [0262], discloses that Figure 2 of Gressel demonstrates a "swap sequence configuration enacted by *randomly alternating the device between one feedback configuration to a second feedback configuration*." Thus, Gressel does not teach anything about "*periodically switching* between iterative outputs generated by two or more linear feedback shift registers," as recited in Claims 4-6. As for Gressel, at paragraphs [0263-0264], Gressel teaches swapping between different feedback configurations to effect pseudorandom number generation. Furthermore, the Examiner alleges that "Par. [0281-0282] of Gressel discloses randomizing the three non-linear feedback shift registers nLFSRs 640, 650 and 660 by using slip trigger generator to switch in a regular interval." While Gressel, at paragraph [0281], discloses slip triggers generated from a slip trigger generator, the use of slip triggers does not teach anything about "*periodically switching* between iterative outputs generated by two or more linear feedback shift registers," as recited in Claims 4-6. As shown in Figure 7, these slip triggers are used in conjunction with a feedback signal of a linear feedback shift register. Therefore, these slip triggers are unrelated to switching between iterative outputs generated by two or more linear feedback shift registers. Thus, based on the foregoing reasons, the Appellants respectfully submit that the Examiner has not shown a teaching of each and every element recited in Claims 4-6. Consequently, Claims 4-6 contain patentable subject matter. The Appellants request a reversal of the rejection to Claims 4-6.

D. Dependent Claims 14-16

Using Claim 14 as exemplary for Claims 14-16, Claim 14 is directed to:

14. The method of Claim 4 further comprising operating a nonlinear operator on said pseudo-random number and one or more operands.

Regarding Claims 14-16, the Examiner's Answer states:

> Appellant's argument regarding Gressel [sic] failure to teach "operating a nonlinear operator on said pseudo-random number generated or output by a linear

> shift register and one or more operands", appeal brief page 12 last paragraph, is not claimed (the italic part is not claimed). **Claim 1 recites generating pseudo-random number by linear feedback shift register and, as cited by the examiner, Gressel par. (0044-0046, 0026 and 0098) discloses a linear feedback shift register generating random numbers.**
>
> Appellant's argument regarding Gressel [sic] failure to teach "operation performed on the pseudo-random number that is generated from a shift register," appeal brief page 12-13 last paragraph, is not persuasive because it is not claimed. Claims 14-16 recite "operating a nonlinear operator on said pseudo-random number and one or more operands." Gressel teaches operating a nonlinear feedback shift register by using a NAND operator to insert a zero operand and NOR operator to insert a one operand into next output (see par. [0217]). Gressel [sic] nLFSR further uses XOR operator (see par. [0239]).
>
> Therefore Gressel does not teach away from a "linear feedback shift register."
>
> *See* Examiner's Answer at page 10.

The Examiner alleges that "Gressel, at paragraph [0217], teaches operating a nonlinear feedback shift register by using a NAND operator to insert a zero operand and NOR operator to insert a one operand into next output.

The Examiner references Gressel, at paragraphs [0044-0046], which states:

> Linear Feedback Shift Registers are linear in the sense that any sequence in the register is followed by another (only one) defined sequence in the register, cyclically, until all sequences have been generated. Non-linear LFSRs can generate more than one sequence from any given sequence.
>
> LFSRs can be configured as in the embodiments presented herein, or equivalently with feedback schemes as suggested in Dixon, R. C., Spread Spectrum Systems, Wiley-Interscience, New York, 1976, Chapter 3, or by table look up devices.
>
> If an adversary or hacker knows 2^n (2 to the power of n) bits of a

sequence of an unmodified n bit LFSR, he or she can easily derive the feedback configuration, which produced the sequence. If an oracle knows the configuration of an unmodified LFSR, and he/she can sample the contents of the device at a given clock cycle, if he/she can know the number of clock cycles that occurred before or after the known clock period, he/she can derive the contents at such given instant. All of the embodiments preferably have elements, which prevent the hacker from estimating the stage of the output at a given sampling, as all embodiments contain non-linear functionality derived from random sources.

Furthermore, the Examiner references Gressel, at paragraph [0026], which states:

> Many mathematical functions, which can be implemented in hardware or software, produce sequences, which pass all tests for randomness for almost all numerical inputs. Such functions are called pseudo-random, since if an observer knew both the input and the function, he could know the "pseudo-random" output. There is no complete randomness, but a number is called random and unpredictable, or intractably difficult to compute, if an observer has insufficient or little knowledge or control of the inner variables of a generator at a given time of sampling, and would have difficulty using his limited knowledge to predict future outputs. Non-predictability means that the output of the feedback shift register is a sampling that has an intractably externally indiscernible correlation to a previous sampling of a plurality of sequence generator outputs and is computationally difficult to predict without knowledge of the internal state of the microelectronic random number string generator.

Furthermore, the Examiner references Gressel, at paragraph [0098], which states:

> Also provided, in accordance with a preferred embodiment of the present invention, is a method for generating a sequence of random numbers including using an nLFSR to generate an nLFSR generated string, operating a random slip actuating triggering process which randomly and without correlation to the LSFR

> generates at least one slip actuating triggers respectively triggering at least one slip generating process, thereby to define a modified string including the nLFSR generated string to which the at least one slip generating processes have been applied, wherein each slip generating process, responsive to occurrence of a slip actuating trigger, reverses the most significant bit of a current number in the nLFSR generated string, and operating a random sampling triggering process which, randomly and without correlation to the nLSFR and without correlation to the random slip actuating triggering process, triggers a sampling of the modified string, thereby to generate a subsequence of the modified string which includes an output string of random numbers.

The Examiner states that the italicized portion of Appellants' argument is unclaimed. In the Brief on Appeal (on page 12), the Appellants intentionally bracketed the italicized portion to indicate to the Examiner that that this portion was not recited in Claim 14. However, based on antecedent basis reasons, in reference to Claim 1 which it depends from, the pseudo-random number *is* generated by a linear shift register (i.e., Claim 1 recites "a method of generating pseudo-random numbers using a linear feedback shift register ..."). Thus, the Examiner's argument is without merit.

Furthermore, the Examiner states that "Claim 1 recites generating pseudo-random number by linear feedback shift register and, as cited by the examiner, Gressel par. (0044-0046, 0026 and 0098) discloses a linear feedback shift register generating random numbers." While Gressel, at paragraphs [0044-0046], discloses using random sources to generate non-linear functionality to a linear feedback shift register, there is no disclosure or teaching of "operating a nonlinear operator on said pseudo-random number and one or more operands," as recited in

Claims 14-16. Thus, for at least this reason, Claims 14-16 contain patentable subject matter. Therefore, the Examiner has not shown a teaching of Claims 14-16.

Furthermore, while Gressel, at paragraph [0026], describes how mathematical functions may be used to produce sequences, Gressel, at paragraph [0026], does not disclose anything about "operating a nonlinear operator on said pseudo-random number and one or more operands," as recited in Claims 14-16. Thus, for at least this reason, Claims 14-16 contain patentable subject matter. Therefore, the Examiner has not shown a teaching of Claims 14-16.

For at least the reason that Gressel, at paragraph [0098], discloses using a *non-linear* feedback shift register (nLFSR), Gressel does not teach "generating pseudo-random numbers using a linear feedback shift register," as recited in Claim 1. Thus, Claim 1 contains patentable subject matter that should be advanced to allowance.

The Examiner references Gressel, at paragraph [0217], which states:

> Nonlinear Feedback Shift Register (nLFSR): Classes of electronic devices wherein the XORed feedbacks from the shift register do not completely determine the sequence of output words. The non-linear methods used in the preferred embodiments, include; a NAND gate to insert a zero into an output sequence when all sensed inputs are one; a NOR gate to insert a one into the next output word, when all sensed inputs are zero; a "slip" pulse which occasionally complements a feedback binary symbol; a control "swap" which alternates the feedback structure thus changing a bit word output sequence.

Contrary to what the Examiner believes, Gressel, at paragraph [0217], discloses how non-linear methods may be used in the preferred embodiments, which include: a NAND gate to insert a zero into an output sequence when all sensed inputs are one; a NOR gate to insert a one into the next output word when all sensed inputs are zero. However, nowhere does Gressel, at paragraph [0217], disclose "operating a nonlinear operator on said pseudo-random number and one or more operands," as recited in Claims 14-16. Thus, based on the foregoing reasons, the Appellants respectfully submit that the Examiner has not shown a teaching of each and every element recited in Claims 14-16. Consequently, Claims 14-16 contain patentable subject matter. The Appellants request a reversal of the rejection to Claims 14-16.

The Examiner further alleges that "Gressel [sic] nLFSR further uses XOR operator (see par. [0239]).

The Examiner references Gressel, at paragraph [0239], which states:

> Slip Sequence Function: A function used in both preferred embodiments that causes a pseudo-random jump displacement in a conventional LFSR. The slip is from one the conventional LFSR sequence to another word in the conventional LFSR sequence. XORing a feedback signal with a random pulse of polarity one implements the process. A slip process preferably is enacted at random intervals occurring a plurality of primary clock cycles more than double the length of the generating nLFSR, to typically avert shortened cyclical sequences.

While Gressel, at paragraph [0239], discloses that a slip sequence function is used to cause a pseudo-random jump displacement in a convention linear feedback shift register. Gressel, at paragraph [0239], further discloses that the slip is generated by XORing a feedback signal with a random pulse of polarity one," which is different from what is recited in Claims 14-16. The Appellants respectfully submits that generating a slip by XORing a feedback signal with a random pulse does not teach "operating a nonlinear operator on said pseudo-random number and one or more operands," as recited in Claims 14-16. Thus, based on the foregoing reasons, the Appellants respectfully submit that the Examiner has not shown a teaching of each and every element recited in Claims 14-16. Consequently, Claims 14-16 contain patentable subject matter. The Appellants request a reversal of the rejection to Claims 14-16.

II. REJECTION OF CLAIM 1 UNDER 35 U.S.C. § 102(e)

The Appellants maintain the arguments previously presented in the Brief on Appeal.

A. Independent Claim 1

Claim 1 is directed to:

1. A method of generating pseudo-random numbers using a linear feedback shift register in which the correlation between successive pseudo-random numbers is reduced, said method comprising sampling output sequences of said linear feedback shift register with a specified periodicity.

Regarding Claim 1, the Examiner's Answer states:

> Appellant's argument regarding Meiyappan failure to disclose "reducing the correlation between successive pseudo-random numbers," Appeal brief pages 15-16, argument is not persuasive because Meiyappan's abstract teaches a generation of truly random numbers using enhanced random number generating technique to achieve unpredictable key security (see also col. 1 lines 19-24). The truly random numbers are generated using the random number generating technique *that includes a method of periodically clocking the output of sampling switch 110* (see col. 3 lines 14-32 and fig. 2 element 206), as the appellant's disclosure discloses, on par. 0028, the use of periodic sampling reduces the correlation between successive outputs of an LFSR, Meiyappan's LFSR 112 also uses periodic sampling and therefore it reduces the correlation between successive outputs of the LFSR.
>
> **Regarding argument, according to Meiyappan's fig. 1, the linear**

> **feedback shift register is simply input into the sampling switch 110, in which the "sampling switch 110 samples (the N bit sample obtained from the Bit Recorder block 108) during periods when its sampling input line is active and does not sample when its sampling input is inactive," appeal brief pages 17-18, argument is not persuasive because as Meiyappan col. 3 lines 1-32 discloses the bit recording [sic] that is based on random N-bit number generated by the LFSR, each bit of the N-bit number may be fed to each of N multiplexers, the selected signal for the multiplexers is derived from the LFSR output bits such that each multiplexer outputs a different bit of the N-bit sample output. Therefore the sampling switch 110 is sampling the output of the LFSR 112 and further the output of sampling switch 110 is periodically clocked see col. 3 lines 29-32. Therefore the N-bit number is generated by the LFSR 112 and outputted to the bit reorder 108 then to the sampling switch as disclosed in col. 3 lines 1-32 or the sampling switch uses N-bit numbers that are directly outputted from LSFR 112, as shown in fig. 1.**
> *See* Examiner's Answer at pages 11-12.

The Appellants respectfully submit that Meiyappan discloses using "additive noise" as a basis for generating a random number (see Meiyappan, at col. 2, lines 24-28).

Furthermore, Meiyappan, at the Summary of the Invention (col. 1, line 51 – col. 2, lines 6), states:

> According to one embodiment of the present invention, truly random numbers are generated with a minimum of extra hardware by taking advantage of the noise inherent in a communication channel. Random numbers can thus be generated without specialized manufacturing requirements and can be incorporated into conventional integrated circuits with minimal additional logic. The random number generation technique offloads the processor from performing extensive random number generation calculations without the use of a hardware accelerator. This random number generation technique may find application in any network device that participates in a virtual private network or is used to

> implement secure electronic commerce.
>
> According to one aspect of the present invention, a method for generating a random value includes: monitoring a signal obtained from a communication channel where the signal includes additive noise, sampling the signal to generate a random value, and storing the random value.
>
> A further understanding of the nature and advantages of the inventions herein may be realized by reference to the remaining portions of the specification and the attached drawings.

Thus, Meiyappan, at col. 1, line 51 – col. 2, lines 6, uses the received demodulated noisy signal to generate a random value.

The Examiner references Meiyappan, at col. 3, lines 1-32, which states:

> At step 204, a bit reordering block 108 psuedo-randomly scrambles the parallel outputs of analog to digital converter 106. This scrambling is performed separately for each N-bit sample output by converter 106. In one embodiment, this reordering is based on a random N-bit number generated by a linear feedback shift register (LFSR) as known in the art (see citation below). For example, each bit of the N-bit number may be fed to each of N multiplexers. The select signal for the multiplexers is derived from the LFSR output bits such that each multiplexer outputs a different bit of the N-bit sample output. This LFSR (not shown) can itself be initialized using a sample from analog to digital converter 106.
>
> Not every N bit sample is used in generating random numbers. At step 206, a sampling switch 110 samples the output of bit reordering block 108. Sampling switch 110 samples during periods when its sampling input line is active and does not sample when its sampling input is inactive. Sampling switch 110 may be implemented by a simple FET. The sampling input to sampling switch 110 is provided by the output of a linear feedback shift register 112. The internal structure of linear feedback shift register 112 is known in the art. Further

> details of linear feedback shift register operation are described in Schneier, Applied Cryptography, (2' d Ed. 1996), pp. 372-378, the contents of this entire volume being incorporated herein by reference for all purposes.
>
> The effect of reordering block 108 and sampling switch 110 is to remove the non-random structure of the transmitted signal and therefore isolate the noise component. The output of sampling switch 110 is also N bits wide and is periodically clocked into a random number storage register 114 at a step 208.

As Meiyappan shows in Figure 1 and explains at col. 2, line 65 – col. 3, line 31, the received signal is demodulated at the analog receiver system and sent to an analog to digital converter. The analog to digital converter outputs bits of the N-bit demodulated signal that is subsequently reordered at a bit reordering block in order to scramble the N bits. The scrambling is performed separately for each N-bit sample output by the analog to digital converter. The reordering is based on a random N-bit number generated by a linear feedback shift register (LFSR) wherein each bit of the N-bit number is fed to each of N multiplexers. The select signal for the multiplexers is derived from the LFSR output bits such that each multiplexer outputs a different bit of the N-bit sample output generated by the analog receiver system. Thus, while the LFSR generates an output that is used to reorder or scramble the N-bit output of the analog to digital converter, Meiyappan does not teach anything about "method of generating pseudo-random numbers using a linear feedback shift register in which the correlation between successive pseudo-random numbers is reduced, said method comprising sampling output sequences of said linear feedback shift register with a specified periodicity," as recited in Claim 1. Furthermore, in particular, Meiyappan does not teach anything about "said method comprising sampling output sequences of said linear feedback shift register with a specified periodicity." Meiyappan's method of generating random numbers is based on reordering (or scrambling) bits generated from an analog to digital converter using the output of a linear feedback shift register (LFSR), in which the output of the LFSR is used to select various bits output by the analog to digital converter. Furthermore, nowhere does Meiyappan disclose anything about *sampling the output sequence of an LFSR with a specified periodicity*. Thus, for at least these reasons, the Examiner has not shown a teaching of Claim 1. Therefore, Claim 1 contains patentable subject matter which should be passed to allowance.

As illustrated in Figure 1 of Meiyappan, the linear feedback shift register is used to provide a sampling input into the sampling switch 110, in which the "sampling switch 110 samples (the N bit sample obtained from the Bit Reorder block 108) during periods when its sampling input line is active and does not sample when its sampling input is inactive." Thus, based on the foregoing information, Meiyappan does not teach anything about "sampling output sequences of said linear feedback shift register with a specified periodicity," as recited in Claim 1. As stated in Meiyappan, at col. 3 lines 14-32, "[t]he sampling input to sampling switch 110 is provided by the output of a linear feedback shift register 112." Thus, Meiyappan says nothing about "sampling output sequences of said linear feedback shift register with a specified periodicity." Meiyappan, at col. 3, lines 29-31, states that "the output of sampling switch 110 is also N bits wide and is *periodically clocked* into a random number storage register 114." While Meiyappan says nothing about periodically sampling an output of a LFSR, Meiyappan merely discloses that the "output of a sampling switch" is periodically clocked (using LFSR output, at each and every output). Thus, Meiyappan does not teach anything about *sampling output sequences of a linear feedback shift register* (since Meiyappan teaches sampling the output of an analog to digital converter using a bit reorder block). Therefore Meiyappan does not teach anything about "sampling output sequences of [a] linear feedback shift register with a specified periodicity," as recited in Claim 1. In other words, Meiyappan is different from what is recited in Claim 1 because the sampling is performed on the output of the analog to digital converter at the bit reorder block 108 by way of using a sampling switch. This has nothing to do with *sampling output sequences of a linear feedback shift register* wherein the sampling is done *with* a *specified periodicity*. Thus, based on the preceding argument, Meiyappan does not teach each and every element and/or feature recited in Claim 1. Consequently, Claim 1 contains patentable subject matter which should be passed to allowance.

III. REJECTION OF CLAIMS 7-10 AND 19 UNDER 35 U.S.C. § 102(e)

The Appellants maintain the arguments previously presented in the Brief on Appeal.

A. Independent Claim 7

Claim 7 is directed to:

7. A method of generating pseudo-random numbers using linear feedback shift registers in which the correlation between successive pseudo-random numbers is reduced, said method comprising periodically switching between iterative outputs generated by at least a first linear feedback shift register and iterative outputs generated by at least a second linear feedback shift register.

Regarding Claim 7, the Examiner's Answer states:

> **Regarding argument since Furuta's "arbitrary number of bits" is shifted after connection of the switching circuit 1309 is switch, Furuta does not teach "periodically switching between iterative outputs generated by at least a first linear feedback shift register and iterative outputs generated by at least a second linear feedback shift register," appeal brief page 20, argument is not persuasive [sic] Furuta col. 67 lines 51-col. 68 lines 2 teaches**

> **switching the switching circuit 1309 in response to the control signal after a predetermined number of bits are shifted in the LFSR 1302. Furuta further provides an example wherein this predetermined number of bits corresponds to the number of bits which are shifted in the LFSR 1302 during one period of the random pulses (periodically switching). Therefore Furuta discloses periodically switching between iterative outputs since switching done by Furuta's LFSR is done at a predetermined time interval for example one period.**
>
> **Regarding argument Furuta failure to teach a "first linear feedback shift register" and a "second linear feedback shift register," appeal brief page 20, argument is not persuasive because col. 67 lines 35-col. 68 line 2 discloses the switching circuit 1309 switching between flip flops 1302(1) and 1302(7) of the random number generator 331. Col. 67 line 49-51 of Furuta discloses the connection of the random number generator 331 shown in Fig. 125 being the same as that shown in Fig. 74. Description for Fig. 74 on col. 44 lines 55-col. 45 lines 5 discloses the flip-flops 1302(1) through 1302(7) are LFSR 1302 registers within the LFSR 331. Therefore there are first LFSR (1302(1)), second LFSR (1302(2)) ... seven LFSR (1302(7)) that are switched using switching circuit 1309 (see fig. 74 and 125) and Furuta does not teach away from what is recited in claim 7.**
>
> *See* Examiner's Answer at pages 12-13.

Furuta, at col. 67, lines 57-59, states that "For example, this predetermined number of bits corresponds to the number of bits which are shifted in the LFSR 1302 during one period of the random pulses." The Appellants respectfully submit that the number of bits that are shifted in the LFSR changes because the period of the random pulses changes. For example, Furuta, at col. 66, lines 49-54 states:

> Therefore, the random number generators 331 shown in FIGS. 74 and 122A through 122C respectively generate bit sequences of the M sequence such

> that the period of the random pulses becomes a maximum, and the random nature of the generated pulses in each period is extremely satisfactory.

As stated in Furuta, at col. 66, lines 49-54, the "random number generators 331" (e.g., element 331, Figure 125) generate bit sequences such that the period of the random pulses becomes a maximum, and the random nature of the generated pulses in each period is extremely satisfactory. Thus, since the random pulses are generated randomly, the number of bits that are shifted would vary. Consequently, for at least this reason, Furuta does not teach "periodically switching between iterative outputs generated by at least a first linear feedback shift register and iterative outputs generated by at least a second linear feedback shift register." Furuta's invention teaches random number generation as controlled by the random nature of pulses that are generated and used by a random number generator.

Contrary to what the Examiner alleges, there is only a single linear feedback shift register pictured in Figure 125. As indicated in Figure 125, element 1302 comprises a linear feedback shift register (LFSR). The LFSR (element 1302) comprises seven flip-flops (denoted 1302_1, 1302_2, 1302_3, 1302_4, 1302_5, 1302_6, and 1302_7). Therefore, Examiner's reference to seven linear feedback shift registers (LFSR) is flawed, as each of these elements represents a flip-flop. Figure 74 represents an embodiment of Figure 125 wherein the switching circuit 1309 is absent. Therefore, neither Figure 74 or 125 discloses "switching between iterative outputs generated by at least a first linear feedback shift register and iterative outputs generated by at least a second linear feedback shift register," as recited in Claim 7.

Furthermore, nothing in Furuta, at col. 67 lines 35-col. 68 line 2, discloses anything about a "first linear feedback shift register" and a "second linear feedback shift register," as recited in Claim 7. Furuta, at col. 67 lines 35-col. 68 line 2, states:

> FIG. 126 shows an embodiment of the switching circuit 1309. This switching circuit 1309 includes an OR gate 1310, AND gates 1311 and 1312, and an inverter 1313 which are connected as shown. The control signal is input to the terminal 1314, and the bits b_0 and b_6 output from the output parts 1303 and 1304 of the flip-flops 1302_1 and 1302_7 are respectively input to terminals 1317 and 1316. An output of the OR gate 1310 is output from a terminal 1318 and is supplied to the input part 1306 of the flip-flop 1302_1 as the output of the switching circuit 1309.
>
> Normally, the switching circuit 1309 selectively outputs the output of the exclusive-OR gate 1305 in response to the control signal. In this case, the connection of the random number generator 331 shown in FIG. 125 is the same as that shown in FIG. 74. But since the random pulses will be repeated periodically if this connection is fixed, this embodiment switches the connection of the switching circuit 1309 in response to the control signal after a predetermined number of bits are shifted in the LFSR 1302.
>
> For example, this predetermined number of bits corresponds to the number of bits which are shifted in the LFSR 1302 during one period of the random pulses. When the connection of the switching circuit 1309 is switched to selectively output the bit b6 from the flip-flop 1302_7, the initial value set in the LFSR 1302 after one period of the random pulses is changed from the original initial value by shifting an arbitrary number of bits in the LFSR 1302. Thereafter, the connection of the switching circuit 1309 is returned to selectively output the output of the exclusive-OR gate 1305. Therefore, it is possible to guarantee the random nature of the random pulses over a plurality of periods of the random pulses.

Thus, Furuta, at col. 67 lines 35-col. 68 line 2, discloses a single linear feedback shift register in which a switching circuit switches between an input from a XOR gate or a flip-flop 1304. Therefore, Furuta does not show a teaching of "said method comprising periodically switching between iterative outputs generated *by at least a first linear feedback shift register and iterative outputs generated by at least a second linear feedback shift register*," as recited in Claim 7.

The Examiner references Furuta, at Figure 74, and at col. 44, lines 55 - col. 45, line 5, which states:

> FIG. 74 shows a first embodiment of the random number generator 331. In FIG. 74, the random number generator 331 includes an exclusive-OR gate 1305 and 7 flip-flops 1302_1 through 1302_7 which are connected as shown. A clock signal is input to a terminal 1307 and is applied to clock terminals CK of each of the flip-flops 1302_1 through 1302_7. The 7 flip-flops 1302_1 through 1302_7 form a 7-bit linear feedback shift register (LFSR) 1302 together with the exclusive-OR gate 1305. An initial value is set in the LFSR 1302, and the LFSR 1302 thereafter repeats a shift operation. As a result, a number from "1" to "127" and excluding "0" is generated once at random within one random number generation period. This random number which is generated from the LFSR 1302 is defined by bits A_1 through A_7, where A_7 denotes the most significant bit (MSB) and A_1 denotes the least significant bit (LSB), for example. However, the bits A_7 and A_1 may respectively denote the LSB and the MSB.

Thus, Furuta, at Figure 74, and at col. 44, lines 55 - col. 45, line 5, further corroborates the fact that elements 1302_1, 1302_2, 1302_3, 1302_4, 1302_5, 1302_6, and 1302_7 correspond to flip-flops used in the creation of a 7 bit linear feedback shift register (LFSR). Thus, Furuta teaches a single linear feedback shift register. For at least these reasons, Furuta does not show a teaching of "switching between iterative outputs generated by at least a first linear feedback shift register and iterative outputs generated by at least a second linear feedback shift register."

B. Dependent Claims 9-10

Taking Claim 9 as exemplary for Claims 9-10, Claim 9 is directed to:

9. The method of Claim 7 wherein said pseudo-random numbers are generated with period equal to the sum of each of the individual linear feedback shift register periods.

Regarding Claims 9-10, the Examiner's Answer states:

> Regarding argument Furuta's "a logical sum of logical products" obtained by using an OR circuit 52 not teaching a "period equal to the sum of each of the individual linear feedback shift register periods," appeal brief page 22 par. 2, argument is not persuasive **because see col. 47 lines 47-59 wherein Furuta teaches the OR circuit 52 that obtains a logical sum of the logical products and outputs equal logical sum. On fig. 79 the signal period sum of logical product input (Yi ∩ Tij) and (Ym ∩ Tmj) for the OR circuit 52 is equal to the signal period logical sum output (logical sum U(Yi ∩ Tij)).**
> *See* Examiner's Answer at page 13.

The Examiner argument references Furuta, at col. 47, lines 47-59, which states:

> One neuron unit 50 receives a plurality of input signals, and a plurality of logical products are obtained between the input signal and the weighting coefficient. Hence, the OR circuit 52 obtains a logical sum of the logical products. Since the input signals are synchronized, the logical sum becomes "111000" when the first logical product is "101000" and the second logical product is "010000", for example. FIG. 79 shows the logical products input to the OR circuit 52 and the 55 logical sum $U(y_i \cap T_{ij})$ which is output from the OR circuit 52. This corresponds to the calculation of the sum and the non-linear function (sigmoid function) in the case of the analog calculation.

The Appellants respectfully submit that Furuta, at col. 47, lines 47-59, describes "a logical sum of logical products" obtained by using an OR circuit 52. As was previously stated in the Brief on Appeal, this does not teach a linear feedback shift register generating pseudo-random numbers "with period equal to the sum of each of the individual linear feedback shift register periods," as recited in Claims 9-10. As illustrated in Furuta, at Figure 20, for example, an OR circuit 52 that is used to logically OR a number of input signals does not teach a "period equal to the sum of each of the individual linear feedback shift register periods." Furthermore, Furuta, at Figure 7, does not teach anything about "wherein said pseudo-random numbers are generated with period equal to the sum of each of the individual linear feedback shift register periods," as recited in Claims 9-10. Furthermore, as stated in the Brief Description of the Drawings, Furuta, at Figure 79, is simply a time chart for explaining the operation of an eighteenth embodiment of the neuron unit according to the present invention. Therefore, Furuta, at Figure 79, does not teach what is recited in Claims 9-10. Consequently, the Examiner has not shown a teaching of each and every element recited in Claims 9-10. Therefore, the Appellants request reversal of the rejections to these claims.

C. Dependent Claim 19

Claim 19 is directed to:

19. The method of Claim 18 wherein said one or more variables comprises the configuration of feedback taps associated with said linear feedback shift register.

Regarding Claim 19, the Examiner's Answer states:

> Regarding argument Furuta col. 44 lines 55-col. 45 lines 32 failure to mention the word "variables" as recited in claim 19 wherein "one or more variables comprises the configuration of feedback taps associated with said linear feedback shift register," appeal brief page 24, is not persuasive because Furuta col. 44 lines 55-col. 45 lines 32 teaches LFSR random number generation using neuron unit that receives input signals (see col. 45 lines 16-32). **The OR circuit of neuron unit 50, see fig. 20 and col. 47 lines 47-col. 48 lines 38, receives two input signals (Yi ∩ Tij) and (Ym ∩ Tmj) (feedback taps) and outputs (logical sum U(Yi ∩ Tij)). Therefore the variables Y, T, i, j, and m comprises the configuration of feedback taps associated with the LFSR 331.**
>
> *See* Examiner's Answer at pages 13-14.

For at least the reason that Claim 19 depends on an allowable Claim 7, Claim 19 should be allowed. Furthermore, for at least the reason that Claim 19 depends on allowable subject matter recited in Claim 18, Claim 19 should be allowed. The Appellants respectfully submit that Examiner's reference to Furuta, at col. 44, lines 55 – col. 45, lines 32, does not teach anything about "wherein said one or more variables comprises the configuration of feedback taps associated with said linear feedback shift register."

The Examiner references Furuta, at col. 44, lines 55 – col. 45, lines 32, which states:

> FIG. 74 shows a first embodiment of the random number generator 331. In FIG. 74, the random number generator 331 includes an exclusive-OR gate 1305 and 7 flip-flops 13021 through 13027 which are connected as shown. A clock signal is input to a terminal 1307 and is applied to clock terminals CK of each of the flip-flops 1302_1 through 1302_7. The 7 flip-flops 1302_1 through 1302_7 form a 7-bit linear feedback shift register (LFSR) 1302 together with the exclusive-OR gate 1305. An initial value is set in the LFSR 1302, and the LFSR 1302 thereafter repeats a shift operation. As a result, a number from "1" to "127" and excluding "0" is generated once at random within one random number generation period. This random number which is generated from the LFSR 1302 is defined by bits

> A_1 through A_7, where A_7 denotes the most significant bit (MSB) and A_1 denotes the least significant bit (LSB), for example, However, the bits A_7 and A_1 may respectively denote the LSB and MSB.
>
> The output of the flip-flop 1032_1 is input to the exclusive-OR gate 1305 in FIG. 74, but the output of any of the flip-flops 1302_1 through 1302_6 may be input to the exclusive-OR gate 1305.
>
> During the backward process, that is, during the learning process, the register 333 stores the weighting coefficient at the time before the learning process is carried out, and the counter 334 is cleared to zero at the time before the learning process is carried out. The error signal pulse sequences which are collected from the neuron unit of the previous stage and are processed in the gate circuit 78 and the frequency dividing circuit 79 shown in FIG. 39, for example, are input as the signals 75 and 76. Hence, a logic operation is carried out based on the signals 75 and 76, the input signal pulse sequence 55, and the random pulse sequence which is output from the comparator 332 based on the weighting coefficient at the time before the learning process is carried out, and a pulse sequence corresponding to the new weighting coefficient is input to the counter 334 via the gate circuit 80. The counter 334 counts the number of pulses of this pulse sequence, and the counted value of the counter 334 is transferred to the register 333 when the learning process ends. As a result, the content of the register 333 is updated or corrected.

Based on what is described in Furuta, at col. 44, lines 55 – col. 45, lines 32, there is no disclosure of "wherein said one or more variables comprises the configuration of feedback taps associated with said linear feedback shift register," as recited in Claim 19. Again, as she did for Claims 9-10, the Examiner references the OR circuit of a neuron unit 50 as illustrated in Furuta, at Figure 20. Figure 20 illustrates various weighting coefficients and input signals received by various AND gates. The output of these AND gates are transmitted to the OR circuit. None of what is disclosed in Figure 20 teaches “wherein said one or more variables comprises the configuration of feedback taps associated with said linear feedback shift register,” as recited in Claim 19. Therefore, Furuta does not teach what is recited in Claim 19. Consequently, the Examiner has not shown a teaching of each and every element recited in Claim 19. Therefore, the Appellants request a reversal of the rejection to Claim 19.

The Examiner references Furuta, at col. 47, lines 47 – col. 48, lines 38, which states:

> One neuron unit 50 receives a plurality of input signals, and a plurality of logical products are obtained between the input signal and the weighting coefficient. Hence, the OR circuit 52 obtains a logical sum of the logical products. Since the input signals are synchronized, the logical sum becomes "111000" when the first logical product is "101000" and the second logical product is "010000", for example. FIG. 79 shows the logical products input to the OR circuit 52 and the logical sum $U(y_i \cap T_{ij})$ which is output from the OR circuit 52. This corresponds to the calculation of the sum and the non-linear function (sigmoid function) in the case of the analog calculation.
>
> When the pulse densities are low, the logical sum of such pulse densities is approximately the sum of the pulse densities. As the pulse densities become higher, the output of the OR circuit 52 saturates and no longer approximates the sum of the pulse densities, that is, the non-linear characteristic begins to show. In the case of the logical sum, the pulse density will not become greater than "1" and will not become smaller than "0". In addition, the logical sum displays a monotonous increase and is approximately the same as the sigmoid function.
>
> As described above, there are two types of couplings (or weighting), namely, the excitatory coupling and the inhibitory coupling. When making numerical calculations, the excitatory and inhibitory couplings are described by positive and negative signs on the weighting coefficient. In this embodiment which uses digital circuits, the couplings are divided into an excitatory group and

> an inhibitory group depending on the positive and negative signs on the weighting coefficient T_{ij}. Then, the calculation up to the part where the logical sum of the logical products of the pulse trains of the input signals and the weighting coefficients are carried out for each group.
>
> In this embodiment, the neuron unit 50 outputs "1" only when the output of the excitatory group is "1" and the output of the inhibitory group is "0" and otherwise outputs "0". This may be achieved by obtaining an AND of a NOT of the output of the inhibitory group and the output of the excitatory group as shown in FIG. 80. Hence, the output a of the excitatory group, the output b of the inhibitory group, and the output y; of the neuron unit 50 can respectively be described by the following formulas (1), (2) and (3).
>
> $a= U (y_i \cap T_{ij(+)})$ (1)
>
> $b= U(y_i \cap T_{ij(-)})$ (2)
>
> $y_i=a \cap b$ (3)
>
> The neural network can be formed by connecting a plurality of such neuron units 50 in a plurality of layers to form a hierarchical structure similarly as in the case of the neural network shown in FIG. 2. When the entire neural network is synchronized, it is possible to carry out the above described calculation in each layer.

While the Examiner alleges that "fig. 20 and col. 47 lines 47-col. 48 lines 38, receives two input signals (Yi ∩ Tij) and (Ym ∩ Tmj) (feedback taps) and outputs (logical sum U(Yi ∩ Tij))," nothing in Furuta, at Figure 20, or at col. 47 lines 47-col. 48 lines 38, teaches "wherein said one or more variables comprises the configuration of feedback taps associated with said linear feedback shift register," as recited in Claim 19. The Examiner alleges that two input signals (Yi ∩ Tij) and (Ym ∩ Tmj) correspond to feedback taps. Instead, Furuta, at col. 47 lines 47-col. 48 lines 38, describes Figure 79 as showing logical products input to a OR circuit 52, and that a logical sum (e.g., (Yi ∩ Tij) and (Ym ∩ Tmj)) corresponds to the calculation of the sum and the non-linear function (sigmoid function) in the case of the analog calculation. Therefore, Furuta does not teach “wherein said one or more variables comprises the configuration of feedback taps associated with said linear feedback shift register,” as recited in Claim 19. Consequently, the Examiner has not shown a teaching of each and every element recited in Claim 19. Therefore, the Appellants request a reversal of the rejection to Claim 19.

IV. REJECTION OF CLAIMS 11-13 UNDER 35 U.S.C. § 102(e)

The Appellants maintain the arguments previously presented in the Brief on Appeal.

A. Independent Claim 11

Claim 11 is directed to:

11. A method of encrypting a pseudo-random number generated by a linear feedback shift register comprising operating a nonlinear operator on said pseudo-random number and one or more operands.

Regarding Claim 11, the Examiner’s Answer states:

> Regarding argument Thomas failure to teach "comprising operating a non linear operator on said pseudo-random number and one or more operands, appeal brief pages 25-27, argument is not persuasive because Thomas teaches operating a non-linear shift registers (operators), to generate a non-linear filtered output, on a random number and one or more operands (i.e. first and second taps) (see par. 0155, 0213 and claim 29).
>
> *See* Examiner’s Answer at page 14.

The Appellants maintain the argument presented in the Brief on Appeal. For the same reasons presented in the Brief on Appeal, the Appellants respectfully submit that Claim 11 is in condition for allowance.

The Examiner references Thomas, at paragraph [0155], which states:

> Referring now to FIG. 10, this figure illustrates a submethod 905 for generating non-linear filtered output bits from shift registers. Step 1005 is the first step of the submethod 905 in which a first tap such as tap 735 and a second tap such as tap 740 of the linear feedback shift register 705 in FIG. 7 are selected. Next, a least significant output bit such as 730 is selected. Next, in Step 1015, the output of the first tap 735 and second tap 740 are combined.

Thomas, at paragraph [0155], discloses "a submethod 905 for generating non-linear filtered output bits from shift registers." A method for generating non-linear filtered output bits does not teach "operating a nonlinear operator on [a] pseudorandom number and one or more operands," as recited in Claim 11. As Thomas clearly states, output bits from shift registers are used. This does not teach "encrypting a pseudo-random number generated by a linear feedback shift register," as recited in Claim 11. Generating nonlinear filtered output bits from shift registers does not teach "operating a nonlinear operator on a pseudo-random number (generated by a linear feedback shift register) and one or more operands." Therefore, Thomas does not teach what is recited in Claim 11. Consequently, the Examiner has not shown a teaching of each and every element recited in Claim 11. Therefore, the Appellants request a reversal of the rejection to Claim 11.

Furthermore, the Examiner references Thomas, at paragraph [0213], which states:

> The present invention has an increased encryption key size that reduces the possibility of a successful attack on a communications channel using the encryption key. The present invention also increases the speed at which a key stream is generated. The present invention generates a key stream that is not

> derived from shift registers possessing linear relationships between feedback taps. The present invention generates a key stream from feedback taps in a non-linear manner which prevents any attacks on the communication channel when the key stream is used to carry information between parties.

Thomas, at paragraph 0213, discloses generating "a key stream from feedback taps in a non-linear manner" which is different from "operating a nonlinear operator on said pseudorandom number and one or more operands." Generating a stream by way of using feedback taps does not teach "operating a nonlinear operator on said pseudo-random number generated by a *linear* feedback shift register and one or more operands." Therefore, Thomas does not teach what is recited in Claim 11. Consequently, the Examiner has not shown a teaching of each and every element recited in Claim 11. Therefore, the Appellants request a reversal of the rejection to Claim 11.

Furthermore, the Examiner references Thomas, at claim 29, which states:

> A system for securing communications channels, comprising:
>
> a register comprising;
>
> a first tap and a second tap for calculating a first value taken between the outputs of the first and second taps, the output between the first tap and second tap comprising a non-linear value;
>
> an output of the register taken between the first value and a third output bit of the register; and
>
> a new bit comprising an operation taken between the taps of the register.

Thomas, at Claim 29, discloses an output between a first tap and a second tap (of a register) comprising a non-linear value. The non-linear value is an intermediary output of a register, which is not a "pseudo-random number generated by a linear feedback shift register," as recited in Claim 11. Furthermore, none of the passages from Thomas disclose anything about "operating a nonlinear operator on said pseudo-random number *and one or more operands*." Therefore, Thomas does not teach what is recited in Claim 11. Consequently, the Examiner has not shown a teaching of each and every element recited in Claim 11. Therefore, the Appellants request a reversal of the rejection to Claim 11.

V. REJECTION OF CLAIM 17 UNDER 35 U.S.C. § 102(e)

The Appellants maintain the arguments previously presented in the Brief on Appeal.

A. Independent Claim 17

Claim 17 is directed to:

17. A method of further encrypting a pseudo-random number generated from a linear feedback shift register by using a hashing function comprising:

receiving said pseudo-random number generated from said linear feedback shift register; and

varying the initial value of said hashing function over time by way of a function operating on one or more variables.

Regarding Claim 17, the Examiner's Answer states:

> A typo is noted on page 27 of the appeal brief line 19. The appellant is trying to disclose "Regarding independent Claim 17, the office action ..." instead of "Regarding independent Claim 11, the office action..."
>
> Regarding argument Walmsley failure [sic] to teach "varying the initial value of said hashing function over time by way of a function operating on one or more variables," appeal brief page 28-29, argument is not persuasive because Walmsley teaches, on par. 0942-0943, and as the office corrected the typo (0942-

0943 not 0942-0934) on page 4 line 10 of the office action mailed on 11/26/2007, the checksum register is used to verify that K1 and K2 (one or more operands) have not been altered by an attacker and the authentication protocol of Walmsley passes the chosen random number (i.e. the initial value) by encrypting both the chosen random number and its digital signature as disclosed on par. [0358-0365] and Walmsley further discloses varying the initial value of the authentication protocol over time since Walmsley uses encrypted time varying random number see par. [0360].

Regarding argument the Walmsley failure to disclose anything about "initial value of said hashing function," appeal brief page 30, argument is not persuasive because the protocol as disclosed on par. [0358-0365] **applies a hash MMAC-SHA-1 as further described on par. [0353-0357, 0771 and 0784]. The checksum register discloses running hash algorithm (SHA-1) on the keys and comparing the result against an internal checksum value, (see par. 1330 and page 41 table 17), therefore the checksum register is applying hash function operating on the keys.**

Regarding argument "initial value of said hashing function" used to "further encrypt a pseudo-random number generated from a linear feed back shift register," appeal brief page 30, argument is not persuasive because it is not claimed. Using the initial value of said hashing function to further encrypt ... is not claimed. The claim preamble recites "A method of further encrypting a pseudo-random number generated from a linear feedback shift register by using a hashing function comprising:" and the preamble is taught by Walmsley (on par. [0338, 0344, and 0358]) the random number is encrypted see par. [0771, and 0774-0775] with a hash function.

See Examiner's Answer at pages 14-15.

The Examiner references Walmsley, at paragraphs [0942-0943], which states:

The Checksum register is a 160-bit number used to verify that K_1 and K_2 have not been altered by an attacker. Checksum is programmed along with K_1, K_2 and R with the authentication chip's SSI (Set Secret Information) command. Since

> Checksum must be kept secret, clients cannot directly read Checksum.
>
> The commands that make use of Checksum are any that make use of K_1 and K_2-namely RND, RD, and TST. Before calculating any revealed value based on K_1 or K_2 a checksum on K_1 and K_2 is calculated and compared against the stored Checksum value. The checksum calculated is the 160-bit value $S[K_1]K_2$].

Furthermore, the Examiner references Walmsley, at paragraphs [0358-0365] which states:

> The protocol passes the chosen random number without the intermediate system knowing its value. This is done by encrypting both the random number and its digital signature.
>
> The protocol has the following advantages:
>
> The secret keys are not revealed during the authentication process. The time varying random number is encrypted, so that it is not revealed during the authentication process.
>
> An attacker cannot build a table of values of the input and output of the encryption process. An attacker cannot call Read without a valid random numbers and signature pair encrypted with the first key. The second key is therefore resistant to a chosen text attack. The random number only advances with a valid call to Test, so the first key is also not susceptible to a chosen text attack.
>
> The system is easy to design, especially in low cost systems such as ink-jet printers, as no encryption or decryption is required by the system itself.
>
> There are a number of well-documented and cryptanalyzed symmetric algorithms to chose from for implementation, including patent-free and license-free solutions.
>
> A wide range of signature functions exists, from message authentication codes to random number sequences to key-based symmetric cryptography.
>
> Signature functions and symmetric encryption algorithms require fewer gates and are easier to verify than asymmetric algorithms.

Appellants maintain the argument presented in the Brief on Appeal. After reviewing Walmsley, at paragraphs [0358-0365] and at [0942-0943], the Appellants do not see how these paragraphs disclose what is recited in Claim 17. Nowhere is there any disclosure of any of the elements and/or features of "varying the *initial value* of said hashing function over time by way of a function operating on one or more variables," as recited in Claim 17. In the Examiner's Answer, the Examiner does not believe that Claim 17 recites an "initial value of said hashing function" used to "further encrypt a pseudo-random number generated from a linear feedback shift register." Clearly, Claim 17 recites a method of "further encrypt a pseudo-random number generated from a linear feedback shift register" in which one of the steps comprises using an "initial value of said hashing function." The Examiner has not shown a teaching of these features recited in Claim 17.

While Walmsley describes a "protocol" that is "able to validate writes and reads of [an] authentication chip's memory space" and a "checksum register" used for verifying checksums against a stored checksum value, nowhere does Walmsley, at paragraphs [0358-0365] and at [0942-0943], disclose "varying the *initial value* of said hashing function over time *by way of a function operating on one or more variables*," as recited in Claim 17. Thus, for each of these reasons, the Appellants respectfully submit that the Examiner has not shown a teaching of what is recited in independent Claim 17. Therefore, the Appellants request the Board to reverse the Examiner's rejection to Claim 17.

The Examiner alleges that a chosen random number teaches the "initial value of a hashing function," as recited in Claim 17. Nowhere does the Examiner substantiate how the chosen random number could possibly teach an "initial value of a hashing function." While

Walmsley, at paragraphs [0358-0365], discloses encrypting both the random number and its digital signature to pass the random number without an intermediate system knowing its value, nowhere does Walmsley disclose anything about "a method of further encrypting a pseudo-random number generated from a linear feedback shift register by using a hashing function comprising: receiving said pseudo-random number generated from said linear feedback shift register; and *varying the initial value of said hashing function over time by way of a function operating on one or more variables*." Therefore, for at least these reasons, the Examiner has not shown a teaching of Claim 17. Thus, the rejection to Claim 17 should be reversed.

Furthermore, the Examiner references Walmsley, at paragraphs [0353-0357], which states:

> The signature function may be held in both chips to generate digital signatures. The digital signature must be long enough to counter the chances of someone generating a random signature. 128 bits is a satisfactory size if S is symmetric encryption, while 160 bits is a satisfactory size if S is HMAC-SHAT.
>
> A test function may be held only in the trusted chip. It may return a value, such as 1, and advance the random number if the untrusted chip is valid; otherwise it may return a value, such as 0, indicating invalidity. The time taken to return a value indicating invalidity must be the same for all bad inputs. The time taken to return the value indicating validity must be the same for all good inputs.
>
> A read function in the untrusted chip may decrypt the random number and signature and then calculate its own signature for the decrypted random number. It may return the data message and a reencrypted random number in combination with the data message if the locally generated signature is the same as the decrypted signature. Otherwise it may return a value indicating failure, such as 0. The time taken to return the value indicating failure must be the same for all bad inputs. The time taken to make a return for a good input must be the same for all good inputs.

> In addition to validating that an authentication chip is present, the protocol is also able to validate writes and reads of the authentication chip's memory space.
>
> The authentication chip's data storage integrity is assumed to be secure-certain parts of memory may be Read Only, others Read/Write, while others are Decrement Only.

Contrary to what the Examiner alleges, Walmsley, at paragraphs [0353-0357], does not disclose anything about applying a "hash MMAC-SHA-1" function. Walmsley, at paragraphs [0353-0357], discloses a "signature function," a "test function," and a "read function," in the authentication of a chip. Therefore, for at least this reason, Walmsley, at paragraphs [0353-0357], does not teach Claim 17. Therefore, for at least these reasons, the Examiner has not shown a teaching of Claim 17. Thus, the rejection to Claim 17 should be reversed.

Furthermore, the Examiner references Walmsley, at paragraph [0771], which states:

> A chosen plaintext attack is not possible against K_1, since there is no way for a caller to modify R, which used as input to the Random function (the only function to provide the result of hashing with K_1).

While the preceding passage from Walmsley, at paragraph [0771], discloses a plaintext attack, the Appellants respectfully submit that none of this verbiage from Walmsley, at paragraph [0771] has anything to do with what is recited in Claim 17. Therefore, for at least these reasons, the Examiner has not shown a teaching of Claim 17. Thus, the rejection to Claim 17 should be reversed.

Furthermore, the Examiner references Walmsley, at paragraph [0784], which states:

> Protocol C1 uses hashing as a form of digital signature. The System sends a number that must be incorporated into the response from a valid authentication

> chip. Since the authentication chip must respond with HMAC [R|M], but has no control over the input value R, the birthday attack is not possible. This is because the message has effectively already been generated and signed. An attacker must instead search for a collision message that hashes to the same value (analogous to finding one person who shares your birthday).

While the preceding passage from Walmsley, at paragraph [0784], discloses hashing as a form of digital signature, the Appellants respectfully submit that none of this verbiage from Walmsley, at paragraph [0784] has anything to do with what is recited in Claim 17. For example, none of this verbiage teaches anything about "varying the initial value of said hashing function over time by way of a function operating on one or more variables." Therefore, for at least these reasons, the Examiner has not shown a teaching of Claim 17. Thus, the rejection to Claim 17 should be reversed.

Furthermore, the Examiner references Walmsley, at paragraph [1330], which states:

> An attacker could theoretically etch off the upper levels of the chip, and deposit enough electrons to change the state of the multi-level Flash memory by 1/3. If the beam is high enough energy it might be possible to focus the electron beam through the Tamper Prevention and Detection Lines. As a result, the authentication chip must perform a validation of the keys before replying to the Random, Test or Random commands. The SHA-1 algorithm must be run on the keys, and the results compared against an internal checksum value. This gives an attacker a 1 in 2 160 chance of tricking the chip, which is the same chance as guessing either of the keys.

While the preceding passage from Walmsley, at paragraph [1330], discloses an authentication chip that performs a validation of keys using an SHA-1 algorithm wherein the results of using this algorithm is compared against an internal checksum value, the Appellants respectfully submit that none of this verbiage from Walmsley, at paragraph [1330] has anything to do with what is recited in Claim 17. For example, none of this verbiage teaches anything about "varying the initial value of said hashing function over time by way of a function operating on one or more variables." Therefore, for at least these reasons, the Examiner has not shown a teaching of Claim 17. Thus, the rejection to Claim 17 should be reversed.

Furthermore, the Examiner references Walmsley, at paragraph [0338], which states:

> Encrypting the random number and the signature using a symmetric encryption function using a first secret key, in the trusted authentication chip;

In addition, the Examiner references Walmsley, at paragraph [0344], which states:

> Encrypting the random number together with the data message by the symmetric encryption function using the second secret key, in the trusted authentication chip;

While the preceding passage from Walmsley, at paragraphs [0338] and [0344], discloses encrypting a random number and a signature by way of a symmetric encryption function using a first secret key and encrypting a random number with a data message by an encryption function using a second secret key, the Appellants respectfully submit that none of this verbiage from Walmsley, at paragraphs [0338] and [0344], has anything to do with "*varying the initial value of said hashing function* over time by way of a function operating on one or more variables." Therefore, for at least these reasons, the Examiner has not shown a teaching of Claim 17. Thus, the rejection to Claim 17 should be reversed.

Furthermore, the Examiner references Walmsley, at paragraph [0358], which states:

> The protocol passes the chosen random number without the intermediate system knowing its value. This is done by encrypting both the random number and its digital signature.

While the preceding passage from Walmsley, at paragraph [0358], discloses encrypting a random number and its digital signature, the Appellants respectfully submit that none of this verbiage from Walmsley, at paragraph [0358] has anything to do with "*varying the initial value of said hashing function* over time by way of a function operating on one or more variables." Therefore, for at least these reasons, the Examiner has not shown a teaching of Claim 17. Thus, the rejection to Claim 17 should be reversed.

Furthermore, the Examiner references Walmsley, at paragraphs [0774-0775], which states:

> The HMAC construct provides security against all forms of chosen plaintext attacks [7]. This is primarily because the HMAC construct has two secret input variables (the result of the original hash, and the secret key). Thus finding collisions in the hash function itself when the input variable is secret is even harder than finding collisions in the plain hash function. This is because the former requires direct access to SHA-1 (not permitted in Protocol Cl) in order to generate pairs of input/output from SHA-1.
>
> The only values that can be collected by an attacker are HMAC[R] and HMAC[RIM]. These are not attacks against the SHA-1 hash function itself, and reduce the attack to a differential cryptanalysis attack (see Section 5.5.13), examining statistical differences between collected data. Given that there is no differential cryptanalysis attack known against SHA-1 or HMAC, Protocol C1 is resistant to the adaptive chosen plaintext attacks. Note that Protocol C3 is not susceptible to this attack.

While the preceding passage from Walmsley, at paragraphs [0774-0775], discloses security against plaintext attacks by way of using secret input variables (the result of the original hash, and the secret key), the Appellants respectfully submit that none of this verbiage from Walmsley, at paragraphs [0774-0775], has anything to do with "*varying the initial value of said hashing function* over time by way of a function operating on one or more variables." Therefore, for at least these reasons, the Examiner has not shown a teaching of Claim 17. Thus, the rejection to Claim 17 should be reversed.

Thus, based on each of the foregoing reasons, the Appellants respectfully submit that the Examiner has not shown a teaching of each and every element recited in Claim 17. Therefore, the rejection to Claim 17 should be reversed.

VI. REJECTION OF CLAIM 18 UNDER 35 U.S.C. § 103(a)

The Appellants maintain the arguments previously presented in the Brief on Appeal.

A. Dependent Claim 18

Claim 18 is directed to:

18. The method of Claim 7 further comprising:

receiving said pseudo-random number generated from said linear feedback shift register; and

varying the initial value of said hashing function over time by way of a function operating on one or more variables.

For at least the reasons provided in Claim 17, the Appellants respectfully submit that Claim 18 contains patentable subject matter. Furthermore, regarding Claim 18, the Examiner's Answer states:

> Regarding argument Gressel failure to disclose "varying the initial value of said hashing function over time by way of a function operating on one or more variables," appeal brief page 31, argument is not persuasive because Gressel par. 0197 discloses using SHA-1 hash function operating on Band N variables and varying the initial value of said hashing function over time by way of a function operating on one or more variables is disclosed in par. [0455] wherein fig. 33 discloses a random number generating device including device of fig. 10, that is a clocked pseudo-random binary number sequence generator, and a secure Hash Standard Coprocessor, that receive the initial output values from the two nLFSRs operative to compress the data into 160 bit ransom strings. **The LSFR of fig. 10 as described on par. [0080] is operative to generate a cyclic output sequence of binary number including a string of binary symbols, the cyclic output sequence including a basic sequence which is generated repeatedly, at least one bit stream generator generating a clocked bit stream including a stream of binary symbols of a first type occasionally interrupted by a binary symbol of a second type, wherein a first varying time interval between the occasional interruptions is intractably correlated to the output sequence of the number sequence generator.**
>
> *See* Examiner's Answer at pages 15-16.

The Examiner references Gressel, at paragraph [0197], which states:

> Hash: A process of converting a larger binary string, typically 10K bits long, divided into blocks 512 or 1024 bits long, processing the result into a much shorter string, typically 128 or 160 bits long. A hash process is typically programmed such that adversaries are unable to replace a valid hashed message with a fraudulent message such that the hashed result might be identical to the valid result. Examples of hash functions are H=B ^ 2 mod N, wherein B is the input, N is a prime number and the hashed result is H. A state of the art secured hash standard is SHA-1.

While Gressel, at paragraph [0197] defines a "coprocessor", and a "hash" respectively, nowhere does Gressel disclose anything about "*varying the initial value* of said hashing function over time by way of a function operating on one or more variables." Therefore, for at least these reasons, the Examiner has not shown a teaching of Claim 18. Thus, the rejection to Claim 18 should be reversed.

Furthermore, the Examiner references Gressel, at paragraph [0455], which states:

> FIG. 33 is a simplified block diagram of a preferred embodiment of a random number generating device. The device includes the device of FIG. 10 and a Secured Hash Standard Coprocessor, operative to receive the output of unprocessed sequences from the two nLFSRs of FIG. 10, operative to compress the data into 160 bit random strings.

The Examiner alleges that Gressel, at paragraph [0455], discloses "varying the initial value of said hashing function over time by way of a function operating on one or more variables," as recited in the second clause of Claim 18. However, Gressel, at paragraph [0455], discloses that Figure 33 is a block diagram of a random number generating device. After reviewing the block diagram shown in Figure 33, the Appellants do not see how Figure 33 teaches anything about "varying the initial value of said hashing function over time by way of a function operating on one or more variables," as recited in the second clause of Claim 18. Therefore, for at least these reasons, the Examiner has not shown a teaching of Claim 18. Thus, the rejection to Claim 18 should be reversed.

The Examiner further alleges that "the LSFR of fig. 10 as described on par. [0080] is operative to generate a cyclic output sequence of binary number including a string of binary symbols, the cyclic output sequence including a basic sequence which is generated repeatedly, at least one bit stream generator generating a clocked bit stream including a stream of binary symbols of a first type occasionally interrupted by a binary symbol of a second type, wherein a first varying time interval between the occasional interruptions is intractably correlated to the output sequence of the number sequence generator."

The Examiner references Gressel, at paragraph [0080], which states:

> There is thus provided, in accordance with a preferred embodiment of the present invention, microelectronic apparatus for generating random binary words including at least one clocked pseudorandom binary number sequence generator normally operative to generate a cyclic output sequence of binary numbers, each number including a string of binary symbols, the cyclic output sequence including a basic sequence which is generated repeatedly, at least one bit stream generator generating a clocked bit stream including a stream of binary symbols of a first type occasionally interrupted by a binary symbol of a second type, wherein a first varying time interval between the occasional interruptions is intractably correlated to the output sequence of the number sequence generator, wherein each occurrence of an interruption of the stream of binary symbols of the first type by a binary symbol of the second type causes a pseudorandom modification of the cyclic output sequence of the number sequence generator, and a sampling device operative to sample the cyclic output sequence of binary numbers thereby to generate a sampled output sequence including at least one sampled binary word.

The Appellants do not see the relevance of the Examiner's statement. The Appellants respectfully submit that the operation of the "pseudorandom binary number sequence generator" described in Gressel, at paragraph [0080], has nothing to do with teaching "varying the initial value of said hashing function over time by way of a function operating on one or more variables," as recited in the second clause of Claim 18. Furthermore, while Figure 10 illustrates the various circuit blocks used to form a pseudorandom binary number sequence generator, Figure 10 does not disclose anything about "varying the initial value of said hashing function over time by way of a function operating on one or more variables." Therefore, for at least these reasons, the Examiner has not shown a teaching of Claim 18. Thus, the Appellants believe that Claim 18 contains patentable subject matter, which should be passed to allowance.

Thus, based on each of the foregoing reasons, the Appellants respectfully submit that the Examiner has not shown a teaching of each and every element recited in Claim 18. Therefore, the rejection to Claim 18 should be reversed.

CONCLUSION

In conclusion, for at least the foregoing reasons provided in this Reply Brief, the Appellants submit that the pending claims are allowable in all respects. Reversal of the Examiner's rejections and issuance of a patent on the Application are therefore requested from the Board.

The Commissioner is hereby authorized to charge additional fee(s) or credit overpayment(s) to the deposit account of McAndrews, Held & Malloy, Account No. 13-0017.

Dated: February 24, 2009

Respectfully submitted,

/Roy B. Rhee/

Roy B. Rhee
Reg. No. 57,303

McAndrews, Held & Malloy, Ltd.
500 West Madison Street, 34th Floor
Chicago, Illinois 60661-2565
Telephone: (312) 775-8246
Facsimile: (312) 775-8100